

Highly Parallel Pipelined VLSI Implementation of Lifting Based 2D Discrete Wavelet Transform

Nikhila Lakshmanan¹, Jayaraj U Kidav², Nandakumar R², and K S Lalmohan²

¹MG University College of Engineering, Thodupuzha, India

Email: nikhi.nikhi27@gmail.com

²National institute of Electronics and Information Technology, Calicut, India

Email: {jayaraj, nanda, lal }@calicut.nielit.in

Abstract— The lifting scheme based Discrete Wavelet Transform is a powerful tool for image processing applications. The lack of disk space during transmission and storage of images pushes the demand for high speed implementation of efficient compression technique. This paper proposes a highly pipelined and distributed VLSI architecture of lifting based 2D DWT with lifting coefficients represented in fixed point [2:14] format. Compared to conventional architectures [11], [13]-[16], the proposed highly pipelined architecture optimizes the design which increases significantly the performance speed. The design raises the operating frequency, at the expense of more hardware area. In this paper, initially a software model of the proposed design was developed using MATLAB®. Corresponding to this software model, an efficient highly parallel pipelined architecture was designed and developed using verilog HDL language and implemented in VIRTEX® 6 (XC6VHX380T) FPGA. Also the design was synthesized on TSMC 0.18μm ASIC Library by using Synopsis Design Compiler. The entire system is suitable for several real time applications.

Index Terms- 2D DWT, Lifting, Parallel pipelining, FPGA, ASIC.

I. INTRODUCTION

For the last two decades, the DWT has gained establishing role in signal processing and image processing applications because of their ability to decompose the signal into different sub bands with both time and frequency information. DWT also has features like progressive image transformation, ease of compressed image manipulation, region of interest coding etc. Earlier DCT was used for image compression applications, but it has several shortcomings such as blocking artifact and bad subjective quality images are restore at high compression ratio. DWT has been traditionally implemented by means of the Mallat filter bank scheme [2]. DWT perform multi resolution analysis which localizes the signals in both frequency and time domain. The blocking artifact at high compression ratio is removed in DWT by its full frame nature which de-correlates the image over large scale. Earlier the implementation of wavelet transform was based on convolution algorithm of filters. But this approach requires a huge amount of computational resources. Hence lifting scheme was used for implementation of DWT. The lifting scheme based DWT has many characteristics, suitable for VLSI hardware implementations. Lifting scheme based DWT provide significant reduction in computational com

plexity and memory usage by providing in-place computation of wavelet coefficients. Lifting scheme has more advantages than convolution algorithm like saving of memory, computational efficiency, integer to integer wavelet transform [10], symmetric forward and inverse transform, etc. With the discovery of CDF (9, 7) filter banks by Cohen, Duabechies [1] and Feauveau, which has linear phase and excellent image compression performance, the application of DWT, has increased tremendously. Various Very Large-Scale Integration (VLSI) architectures of the DWT have presented in the literature [3]-[7].

Several lifting based DWT architectures have been developed. Wu et al [11] have proposed a line based scanning schemes and a folded architecture. This architecture performs multilevel DWT and has simple control circuits. But it uses an external frame buffer. Recursive architecture [12] eliminates the requirement of external buffer, but it has a complex control circuit and requires more internal memory than folded structure. All these architectures works at a fixed processing speed, and hence cannot be extended to achieve higher processing speed. Higher processing speed can be achieved with parallel FIR structure, at the expense of more hardware area.

The aim of this paper was to construct a highly pipelined and distributed VLSI architecture for CDF (9,7) lifting based 2D DWT which meets high processing speed requirement, simple control signals and controlled increase of hardware cost. High speed processing was achieved by using techniques such as pipelining, distributed arithmetic, etc. The paper proposes the FDWT architecture for large throughput. The fixed point Q[2:14] format was used to represent the lifting coefficients which reduce the computation complexity and also bring a great advantage to VLSI hardware implementation.

II. LIFTING SCHEME BASED DISCRETE WAVELETTRANSFORM

Lifting scheme based discrete wavelet transform also called as the second generation wavelet, was introduced by Sweldens [8], [9] is based entirely on the spacial method.

A. Lifting Scheme realization

The lifting algorithm can be computed in three main phases, namely: the split phase, the lifting phase and the normalization phase, as illustrated in Figure.1. The lifting scheme algorithm can be described as follow:

- *Split phase*: The original signal, $X(n)$, is split into odd and even samples.

$$X_e = X(2n) \quad (1)$$

$$X_o = X(2n+1) \quad (2)$$

- *Lifting phase*: This phase is executed as N sub-steps, where the odd and even samples are filtered by the prediction and update filters, $P_n(n)$ and $U_n(n)$.

In this phase the even samples are multiplied by predictor operator and are used to predict the odd samples. The difference between the odd sample and the predicted value gives the detail coefficient. Then the even samples are updated with detail coefficients to get smooth coefficients.

$$Y(2n+1) = X_o(2n+1) + P_n(X_e) \quad (3)$$

$$Y(2n) = y(2n+1) + U_n(X_e) \quad (4)$$

(4) - *Normalization or Scaling step*: After N lifting steps, a scaling coefficients K and $1/K$ are applied respectively to the odd and even samples in order to obtain the low-pass coefficients ($Y_L(i)$), and the high-pass coefficients ($Y_H(i)$).

$$Y_L(i) = K * Y(2n) \quad (5)$$

$$Y_H(i) = 1/K * Y(2n+1) \quad (6)$$

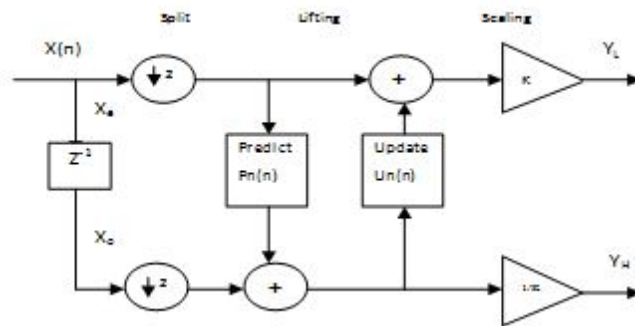


Figure 1. General diagram of the lifting process

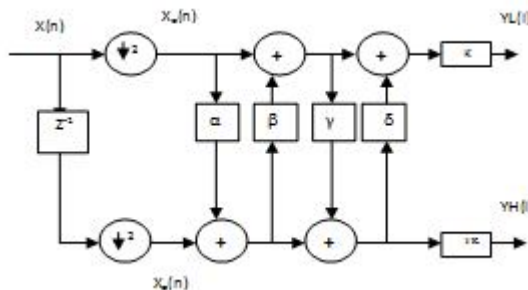


Figure 2. CDF(9,7) lifting filter architecture

The CDF(9,7) wavelet filters is shown in Figure.2. The architecture consists of two lifting stages, where $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$, $\hat{\delta}$ are called the four lifting coefficients and K is the scaling factor or normalization constant. From [9] the lifting and scaling coefficients can be expressed as:

$$\alpha = -1.586134342,$$

$$\beta = -0.0529801185,$$

$$\gamma = 0.882911076,$$

$$\delta = -0.443506852,$$

$$\text{and } K = 1.149604398$$

The lifting scheme algorithm to the (9,7) filter is applied as:

Split step:

$$X_e \leftarrow X(2i) \text{ Even Samples}$$

$$X_o \leftarrow X(2i+1) \text{ Odd Samples}$$

Lifting Steps:

For (9, 7) filter, number of lifting stages, $N=2$

$$\text{Predict P1: } D_1(i) = X_o(i) + \alpha [X_e(i) + X_e(i+1)]$$

$$\text{Update U1: } S_1(i) = X_e(i) + \beta [D_1(i-1) + D_1(i)]$$

$$\text{Predict P2: } D_2(i) = D_1(i) + \gamma [S_1(i) + S_1(i+1)]$$

$$\text{Update U2: } S_2(i) = S_1(i) + \delta [D_2(i-1) + D_2(i)]$$

Scaling Step:

$$Y_H(i) = 1/K * D_2(i)$$

$$Y_L(i) = K * S_2(i)$$

B. Two-Dimensional Discrete Wavelet Transform

The basic idea of 2-D architecture is similar to 1-D architecture. A 2-D DWT can be seen as a 1-D wavelet transform along the rows and then a 1-D wavelet transform along the columns. The 2-D DWT operates in a straightforward manner by inserting array transposition between the two 1-D DWT. The rows of the array are processed first with only one level of decomposition. This essentially divides the array into two vertical halves, with the first half storing the average coefficients, while the second vertical half stores the detail coefficients. This process is repeated again with the columns, resulting in four sub-bands within the array defined by filter output. The LL sub-band represents an approximation of the original image. The other three sub-bands HL, LH, and HH contain higher frequency detail information (mostly local discontinuities in the edges of the image). This process is repeated for as many levels of decomposition as are desired.

III. HARDWARE IMPLIMENTATION

The 2D DWT has a fundamental role in compression algorithms. However, because of its complexity in hardware implementation, a significant number of studies have been devoted to the design of architecture that effectively utilizes the available resources.

A. Proposed 2D DWT System Architecture

In this work a highly pipelined lifting scheme based 2D FDWT has been implemented. Acceleration in design has been achieved through parallel processing of independent sub modules. Figure3 shows the proposed highly pipelined lifting based 2D DWT architecture.

The proposed architecture consist of the following units: stage1 DWT processor, stage2 DWT processor, DWT controller, tile memory, even memory, odd memory, combined memory, transpose memory and two scaled and detail memories. DWT controller was used to control the overall process and was designed as a finite state machine with simple control signals. All the memory modules used in this design are dual ported RAM (DPRAM), that allows multiple read or write to occur at the same time. Each of these memories has two banks (bank 1 & bank 2). The common clock and reset was given to each module in order to maintain synchronism.

The data flow between different modules is explained in

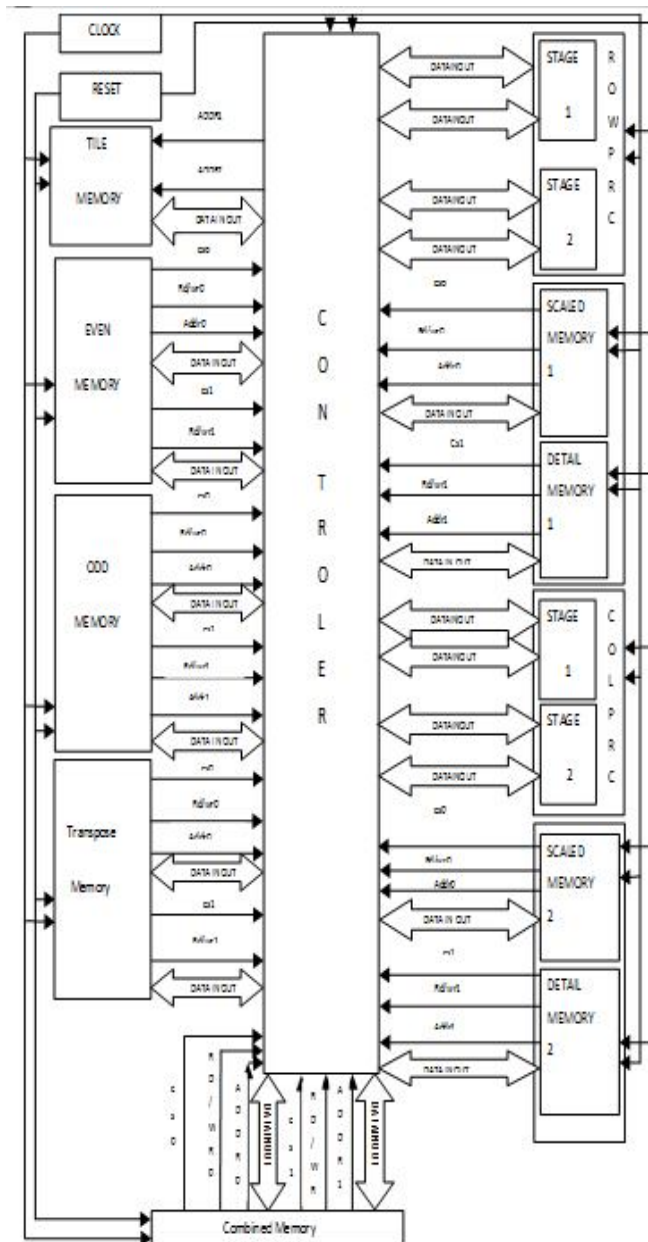


Figure 3. Proposed 2D DWT architecture

detail in this section. Initially the input image coefficients are stored linearly in the external memory. The controller reads the external memory tile wise and writes to bank 1 of tile memory. After writing the first tile data into tile memory, the controller will take the first tile data row wise and split them into even and odd image coefficients and store them in the bank1 of even and odd memories. Then the controller addresses these coefficients to stage1 DWT processor and addresses the transformed coefficients, i.e. the scaled and detail coefficients to bank1 of first scaled and detail memories. Then the controller addresses these stage1 transformed coefficients to stage2 DWT processor and addresses the stage2 transformed coefficients, i.e. the scaled and detail coefficients to bank 1 of second scaled and detail memories. Then this process is repeated for each row in the entire tile image, and finally the result of each consecutive rows are saved in the bank 1 of second scaled and detail memory. By

this consecutive stage1 and stage2 operation on tile image we will get the row wise decomposed coefficients. After the row wise decomposition, the scaled and detail coefficients are assembled together for column wise decomposition. Once the row processing has been complete on the first tile, the results of stage2 DWT processor was combined to form a 2D array and is stored in bank 1 of combined memory. For column processing the combined 2D array was transposed and stored in bank1 of transpose memory. From the transpose memory again the data was split as even and odd coefficients, stored in bank1 of even and odd memories and the process of stage1 and stage2 are repeated until column processing was done on the first tile. When the column processing of first tile is being done, row processing on the next tile will be done in parallel. The next tile will be written into bank2 of tile memory. The controller will take this data row wise, split them as even and odd coefficients and are stored in bank 2 of even and odd memories. These even and odd coefficients are given as to input to stage1 DWT processor. The result of stage1 processing will be stored in bank 2 of first scaled and detail memory. Then the stage2 processing was done on the result of stage1 processing. Once the row wise processing of this tile is complete, the result was given for column operation. The column processing this tile will be done in parallel with row operation on next tile. Hence we achieve high level pipelining between the row and column processing. The above operations are repeated for entire image tile wise. Finally the scaled coefficients on the lowest frequency sub band (LL) of each tile image are assembled to form a 2D array of level1 compressed image. The same routine was again applied on this lowest frequency sub band to get level2 compressed image.

B. Stage1 and Stage2 DWT Processor

This module was mainly used for transformation of image. In this process image are transformed and hence the detail (high pass) and scaled (low pass) coefficients are generated. The stage1 DWT processor consist of registers, adders and multipliers as shown in figure4 and the stage2 DWT processor consist of registers, adders, multipliers and multiplexers as shown in figure5. The input data are 16 bit each. The input data from tile memory was taken row wise, divided into even and odd data and stored in even and odd memories. The input to the stage1 processor comes from these even and odd memories. The outputs from the stage1 DWT processor are store in first scaled and detail memory. Stage1 DWT processor outputs will be given as input to stage2 DWT processor and the output of stage2 DWT processor will be store in second scaled and detail memory row wise. The output of stage 1 and stage2 DWT processors are the scaled and detail coefficients.

For stage1 DWT processor the pixel of the image are given as input. Data from the even memory and odd memory was used for scaled and detail coefficient generation. Initially the current even pixel value and next even pixel value are added and in turn given to multiplication process with filter coefficient. Finally the detail coefficients are achieved from

the addition process of multiplied output and odd pixel value. Again these detail coefficients are taken and added with its previous value and in turn given to multiplication process with filter coefficient. The resultant is added with even pixel value which gives the scaled coefficients. Hence all the values from even and odd memory will be taken and this process will be repeated in order to achieve the scaled and detail coefficients of the entire row.

Now these scaled and detail coefficients were taken as input for the further process. Hence for the stage2 DWT processor, these scaled and detail coefficients are taken as input and will do the stage2 processing in order to obtain the scaled and detail coefficients from the transformed coefficient of stage1 DWT processor. In stage2 DWT processor the same process as in stage1 DWT processor is done, but here the input data are taken from the first scaled and detail memory row wise and finally the obtained scaled and detail coefficients are multiplied by filter gain for normalization.

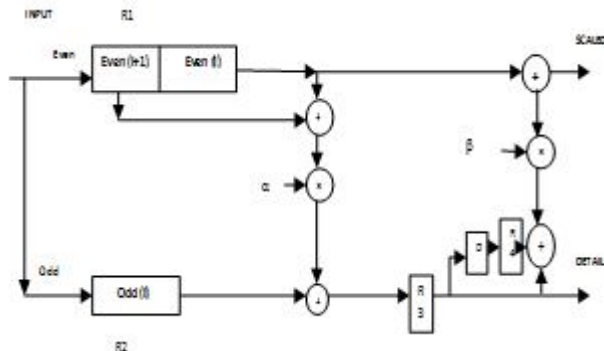


Figure 4. Stage1 2D DWT processor

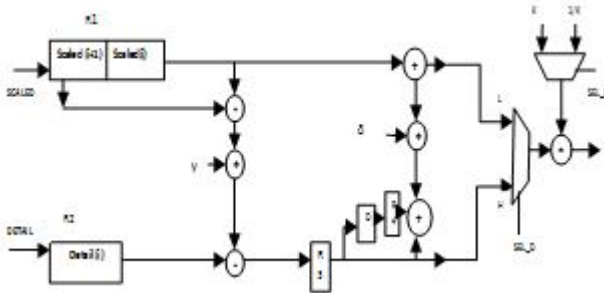


Figure 5. Stage2 2D DWT processor

The lifting coefficients are stored in fixed point Q[2:14] format. Table I shows the fixed point representation of the CDF (9,7) lifting coefficients.

TABLE I. FIXED POINT REPRESENTATION OF LIFTING COEFFICIENTS

Co-efficient	Original value	Q[2:14] representation
α	-1.586134342	-25987
β	-0.0529801185	-868
γ	0.882911076	14465
δ	-0.443506852	7266
K	1.149604398	18835
1/K	0.869864452	14252

C. Highly Parallel Pipelined Image Processing

When the whole image is to be processed, the pixel values are divided into N number of 256x256 size tiles. These tiles

are processed one after the other. Each tile was processed in two steps. First row wise processing was done and then column wise processing. Within the row wise and column wise processing, stage1 and stage2 processing is done row wise. If processing of second tile was done only after complete processing of first tile, then more clock cycles will be consumed. Highly parallel pipelined processing is applied here as a solution to this problem. Then while column processing of first tile is being done, row processing of next tile will be done in parallel manner so that as the next step column processing of the next tile can be done with row processing of the tile after it. Also the stage1 and stage2 processing of each tile was done row wise, i.e. while the stage2 processing of first row within the tile is being done, stage1 processing of next row will be done in parallel manner so that as the next step stage2 processing of the next row can be done with stage1 processing of the row after it. The processor fetch and process the row values of tile N+1 while it process the column values of tile N. Thus the column processing of tile N and row processing of tile N+1 will be completed at the same time. Without pipelining, if there were N tiles, 2N steps are required for the complete image processing. This can be reduced to N+1 step if pipelining is employed. Thus the whole image can be processed in a much reduced time.

IV. IMPLEMENTATION RESULTS

A software model for the proposed 2D DWT was developed, implemented and simulated using MATLAB. Based on this software model, an efficient highly pipelined hardware architecture for the 2D DWT processor was developed in verilog hardware descriptive language and synthesized for VERTEX 6 (XC6VHX380T) FPGA and ASIC technologies. The proposed architecture is in the class of the fastest implementation, because of its highly parallel pipeline processing that limits the critical path delay. The proposed system use large number of memories because of the introduced highly pipelined processing. Compared with the similar architectures, computation time of proposed architecture was reduced with controlled increase of hardware cost. Here Modelsim tool was used in order to simulate and check the functionality of the design. Once the functional verification was done, the design was taken to XILINX tool for synthesis process. The DWT schematic with basic inputs and outputs are shown in figure 6.



Figure 6. DWT schematic with basic inputs and outputs

D. Performance Comparison

Table II shows the performance in terms of processing speed of the proposed system on a 512x512 image.

TABLE II. PROCESSING SPEED OF THE DWT SYSTEM

Clock period= 6.02 ns(Frequency=166MHz)	
Process	Time taken(ns)
Tile 1- Row1- Stage1 Processing	616
Tile 1- Row1- Stage2 Processing and Tile1- Row2 - Stage1 processing	616
Tile 1- Row Processing	158312
Tile 1- Column Processing and Tile 2- Row processing	158312
Total time for processing 512x512 image	791560

A 512x512 image has four 256x256 tiles. The time taken for processing one tile will be 0.16ms. Hence for highly parallel pipelined processing the total time taken will be 5(4+1) times the processing time for one tile, which will be equal to 0.79ms. Hence the proposed highly parallel pipelined architecture helps to achieve better performance for high resolution images like HD (1920x1080) and 720 P image (1280 x720). The performance comparison of several 2D DWT architecture for CDF (9,7) with the proposed architecture is shown in table III.

TABLE III. COMPARISON OF SEVERAL 2D DWT ARCHITECTURE FOR CDF (9,7)

Architecture	Multiplier	Adder	DWT scheme
Chrysafis[13]	32	28	Convolution
Wu[11]	32	32	Convolution
Barua[14]	10	16	Lifting
Cheng[15]	24	76	FIR
Dillen[16]	16	24	Lifting
Proposed	10	16	Lifting

Table IV depicts the design synthesis results for FPGA and table5 depicts the design synthesis results for ASIC technologies. The presented result was obtained for fixed point [2.14] representation of lifting coefficients.

TABLE IV. DESIGN SYNTHESIS RESULTS FOR FPGA TECHNOLOGY

Technology	Xilinx Vertex-6	
Area utilization	Slice LUTs	985
	Slice registers	506
Memory	1027kB	
DSP blocks	15	
Maximum frequency	166 MHz	

TABLE V. DESIGN SYNTHESIS RESULTS FOR ASIC TECHNOLOGY

Technology	TSMC 0.18 μ m
Combinational area utilization	21453
Sequential area utilization	5378
Gate count estimation	9.58 K
Maximum frequency	193 MHz

V. CONCLUSION AND FUTURE WORK

An efficient highly pipelined VLSI architecture design of 2D bi-orthogonal CDF 9/7 2D DWT is proposed in this paper. In this work we have successfully implemented the proposed highly pipelined VLSI architecture on Xilinx FPGA and TSMC 0.18 μ m. The proposed design is suitable for variety of hardware implementation to meet the different processing speed requirement with controlled increase in hardware cost.

Applied highly parallel pipeline processing reduces the critical path delay. The design works with an estimated frequency of 166MHz for Xilinx Virtex6 FPGA and 193 MHz for TSMC 0.18 μ m ASIC Library.

REFERENCES

- [1] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Commun. Pure Appl. Math., vol. 41, pp. 909–996, 1988.
- [2] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet representation," IEEE Trans. Pattern Anal. Mach. Intell. 11, pp. 674–693, 1989.
- [3] G. Knowles, "VLSI Architectures for the Discrete Transform," Electronics Letters, vol. 26, no. 15, 1990, pp. 1184–1185.
- [4] A. S. Lewis, and G. Knowles, "VLSI Architecture for 2-D Daubechies Wavelet Transform Without Multipliers," Electronics Letter, vol. 27, no. 2, pp. 171-173, Jan. 1991.
- [5] K. K. Parhi, and T. Nishitani, "VLSI Architectures for Discrete Wavelete Transforms," IEEE Trans. on VLSI Systems, vol. 1, no. 2, pp. 191-202, June 1993.
- [6] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI Architectures for the Discrete Wavelet Transform," IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 42, no. 5, May 1995.
- [7] Pei-Yin Chen, "VLSI Implementation for One-Dimensional Multilevel Lifting-Based Wavelet Transform," IEEE Transactions on Computers, vol.53 no.4, pp.386-398, April 2004.
- [8] W. Sweldens, "The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions," in Proc. SPIE, 2569, pp.68-79, 1995.
- [9] I. Daubechies, and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," steps', J. Fourier Anal. Appl., vol. 4, no. 3, pp. 247– 269, 1998.
- [10] A.R. Calderbank, I. Daubechies, W. Sweldens, and B.L. Yeo, "Wavelet Transform that Map Integers to Integers," Applied and Computational Harmonic Analysis (ACHA), vol.1.5, no.3, pp. 332-369, 1998.
- [11] Wu, P.C. and Chen, L.G. "An Efficient Architecture for Two-Dimensional Discrete Wavelet transform," IEEE Trans. Circuits Syst. Video Technol., Vol. 11, No. 4, pp. 536–545.
- [12] Huang, C.T. Tseng, P.C. and Chen, L.G. "Generic RAM-based Architectures for two-Dimensional Discrete Wavelet Transform with Line based method," IEEE Trans. Circuits Syst. Video Technol., Vol. 15, No. 7, pp. 910–920, 2005.
- [13] Chrysafis C. and Ortega A., "Line-Based, Reduced Memory, wavelet Image Compression," IEEE Trans. Image Processing., vol. 9, no. 3, pp. 378-389, Mar.2000.
- [14] Barua, S. Carletta J.E., Kotteri K.A., and Bell A.E., "An Efficient Architecture for Lifting-Based Two-Dimensional Discrete Wavelet Transform, "Integration, the VLSI J., vol. 38, no. 3, pp. 341-352, 2005.
- [15] Cheng C. and Parhi K.K., "High-Speed VLSI Implementation of 2D Discrete Wavelet Transform," IEEE Trans. Signal Processing, vol. 56, no. 1, pp. 393-403, Jan. 2008.
- [16] Dillen G. et al., "Combined Line-Based Architecture for the 5-3 and 9-7 Wavelet Transform of JPEG2000," IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 9, pp. 944-950, Sept. 2003.